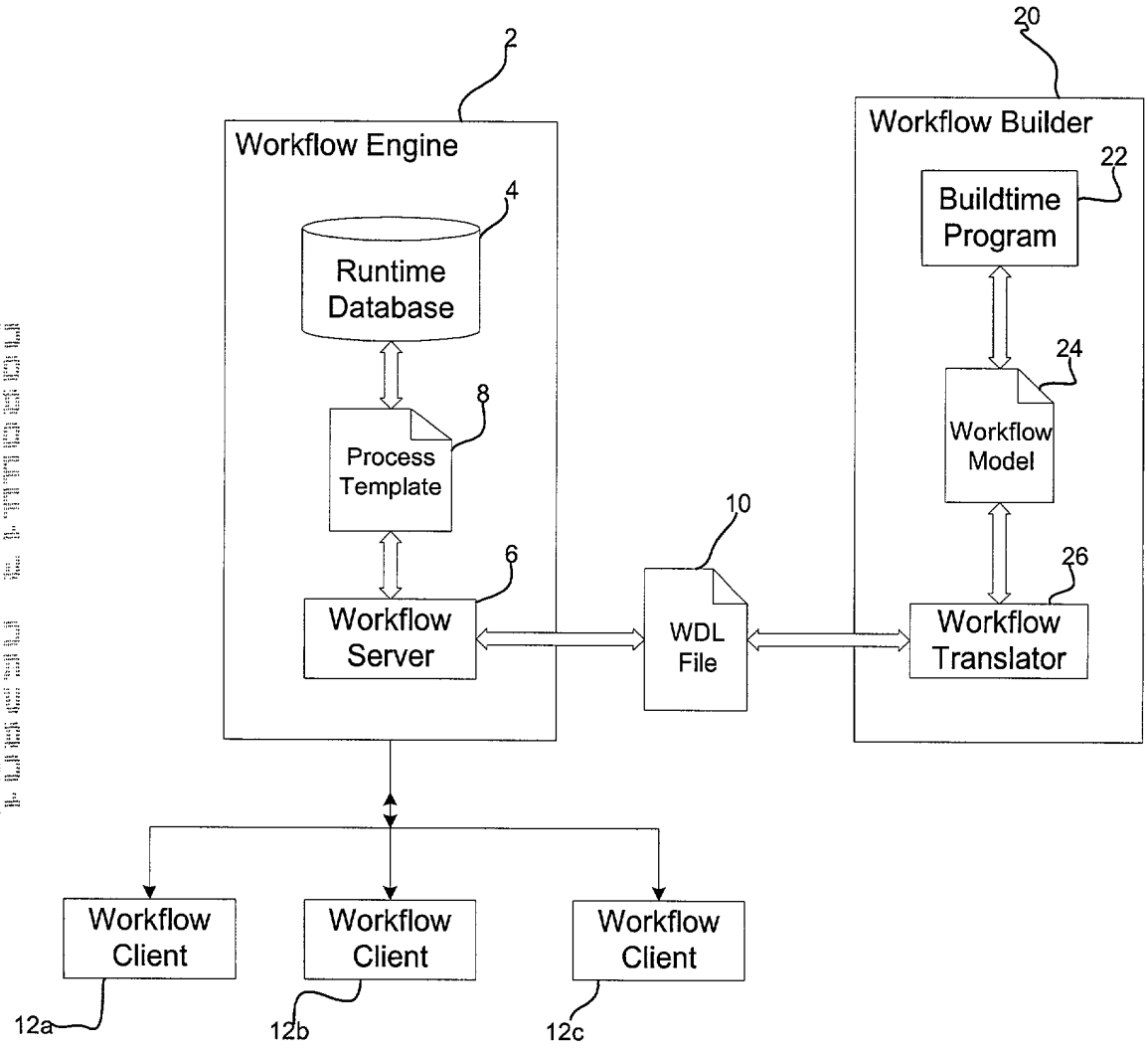
FIG. 1


2
Workflow Engine
4
Runtime Database
8
Process Template
6
Workflow Server
20
Workflow Builder
22
Buildtime Program
24
Workflow Model
26
Workflow Translator
10
WDL File
Workflow Client
Workflow Client
Workflow Client
12a
12b
12c

STL920000095US1
A. Yaung
Sheet 2/12

FIG. 2

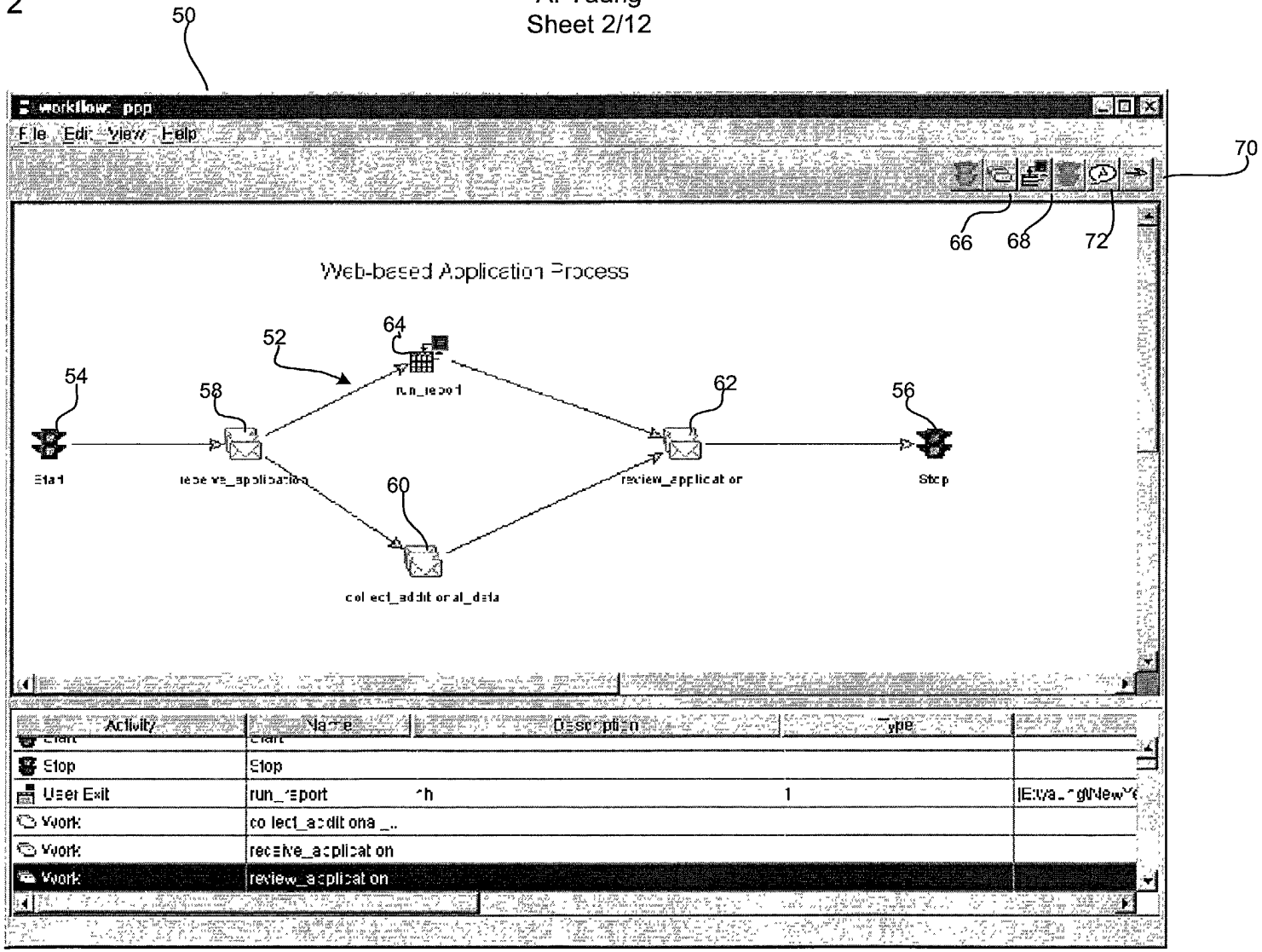


FIG. 3

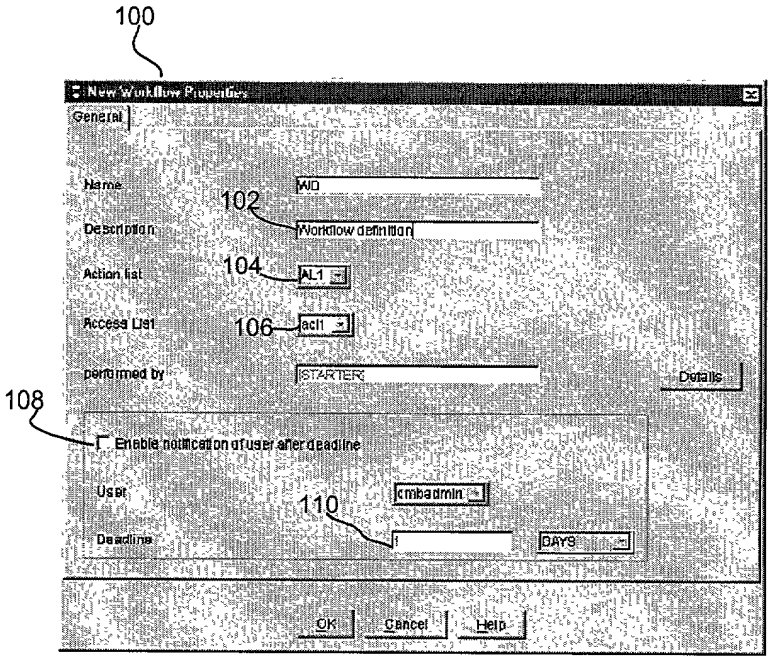

100
New Workflow Properties
General
Name
WD
102
Description
Workflow definition
Action list
104
AL1
Access List
106
acl1
performed by
STARTER
Details
108
Enable notification of user after deadline
User
cmbadmin
110
Deadline
1
DAYS
OK
Cancel
Help

FIG. 4

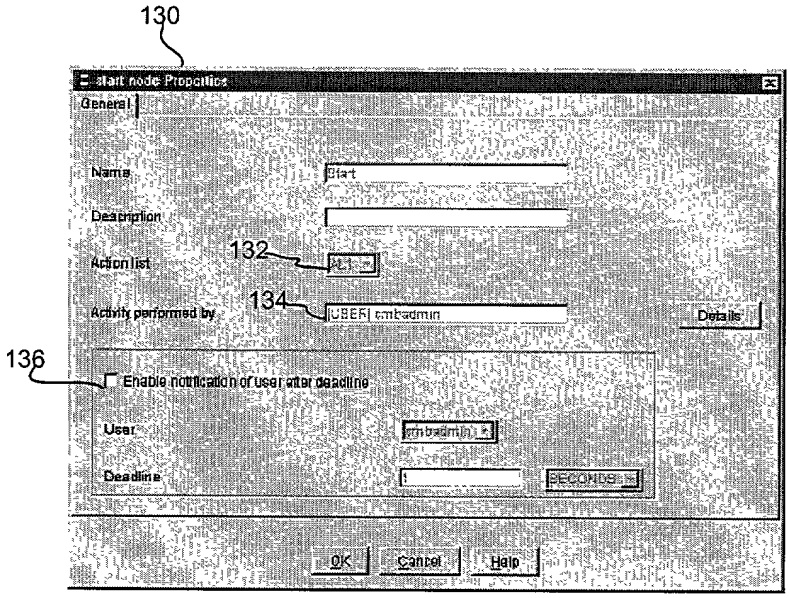
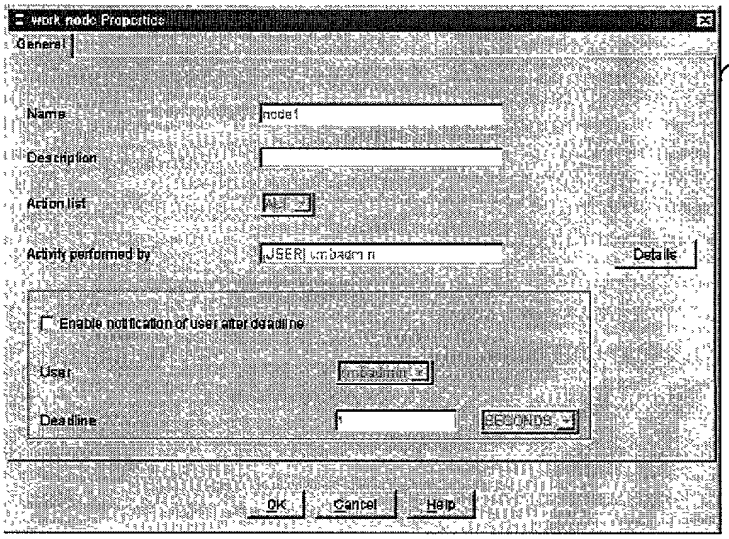

130
start node Properties
General
Name
Start
Description
Action list
132
AL1
Activity performed by
134
USER cmbadmin
Details
136
Enable notification of user after deadline
User
cmbadmin
Deadline
1
SECONDS
OK
Cancel
Help

FIG. 5



150

FIG. 6

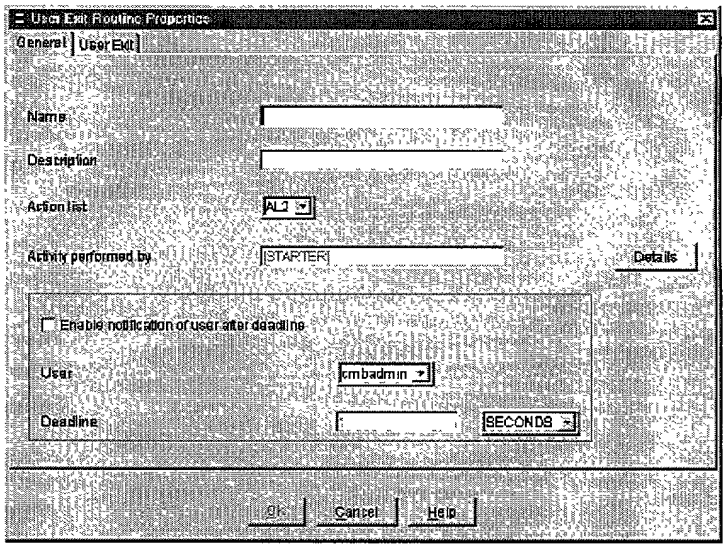
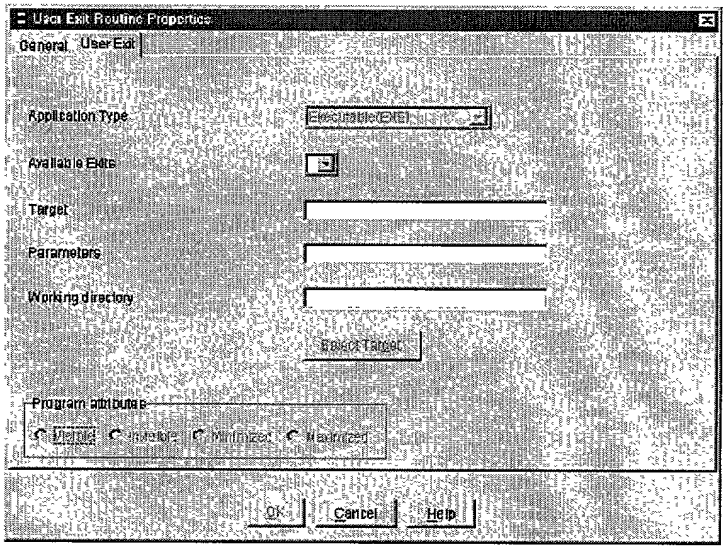


FIG. 7


Uscx Exit Routine Properties
General
User Exit
Application Type
Executable(EXE)
Available Exits
Target
Parameters
Working directory
Browse Target...
Program attributes
Normal
Invisible
Minimized
Maximized
OK
Cancel
Help

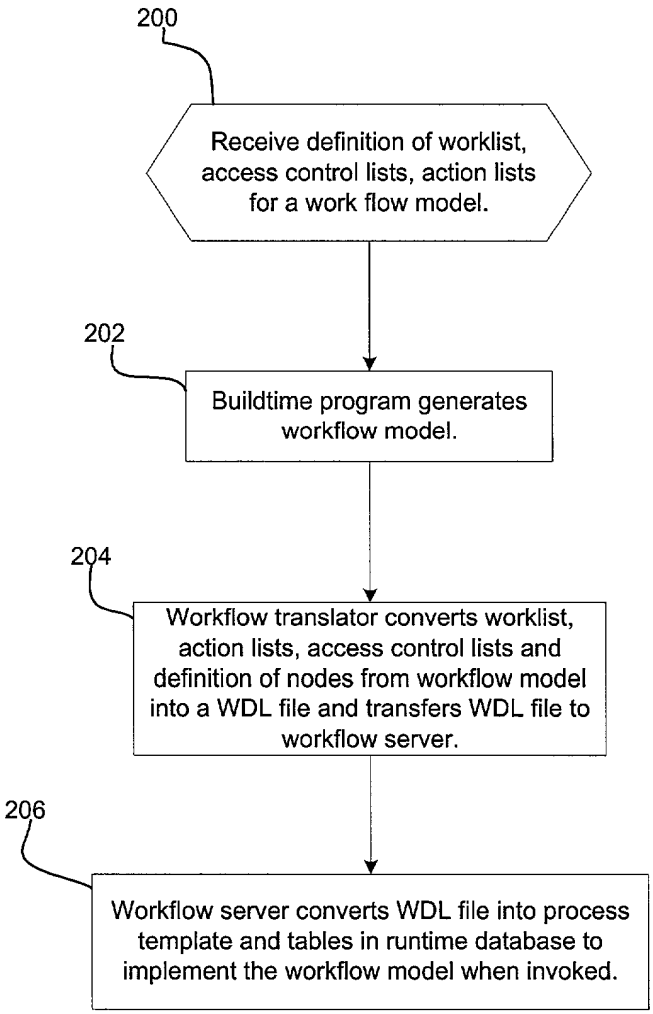

200
Receive definition of worklist, access control lists, action lists for a work flow model.
202
Buildtime program generates workflow model.
204
Workflow translator converts worklist, action lists, access control lists and definition of nodes from workflow model into a WDL file and transfers WDL file to workflow server.
206
Workflow server converts WDL file into process template and tables in runtime database to implement the workflow model when invoked.

FIG. 8

FIG. 9

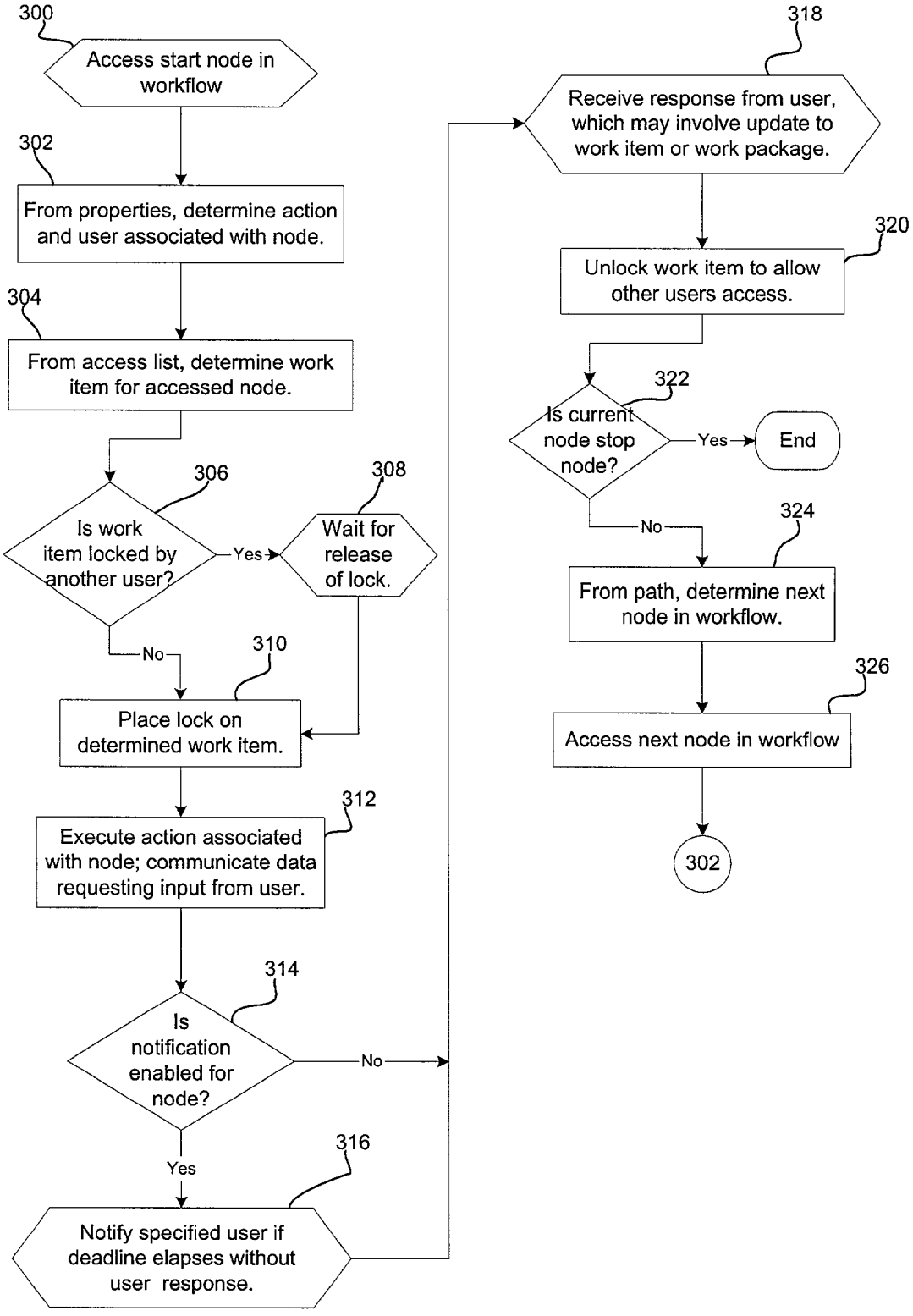

300
Access start node in workflow
302
From properties, determine action and user associated with node.
304
From access list, determine work item for accessed node.
306
Is work item locked by another user?
Yes
308
Wait for release of lock.
No
310
Place lock on determined work item.
312
Execute action associated with node; communicate data requesting input from user.
314
Is notification enabled for node?
No
Yes
316
Notify specified user if deadline elapses without user response.
318
Receive response from user, which may involve update to work item or work package.
320
Unlock work item to allow other users access.
322
Is current node stop node?
Yes
End
No
324
From path, determine next node in workflow.
326
Access next node in workflow
302

FIG. 10

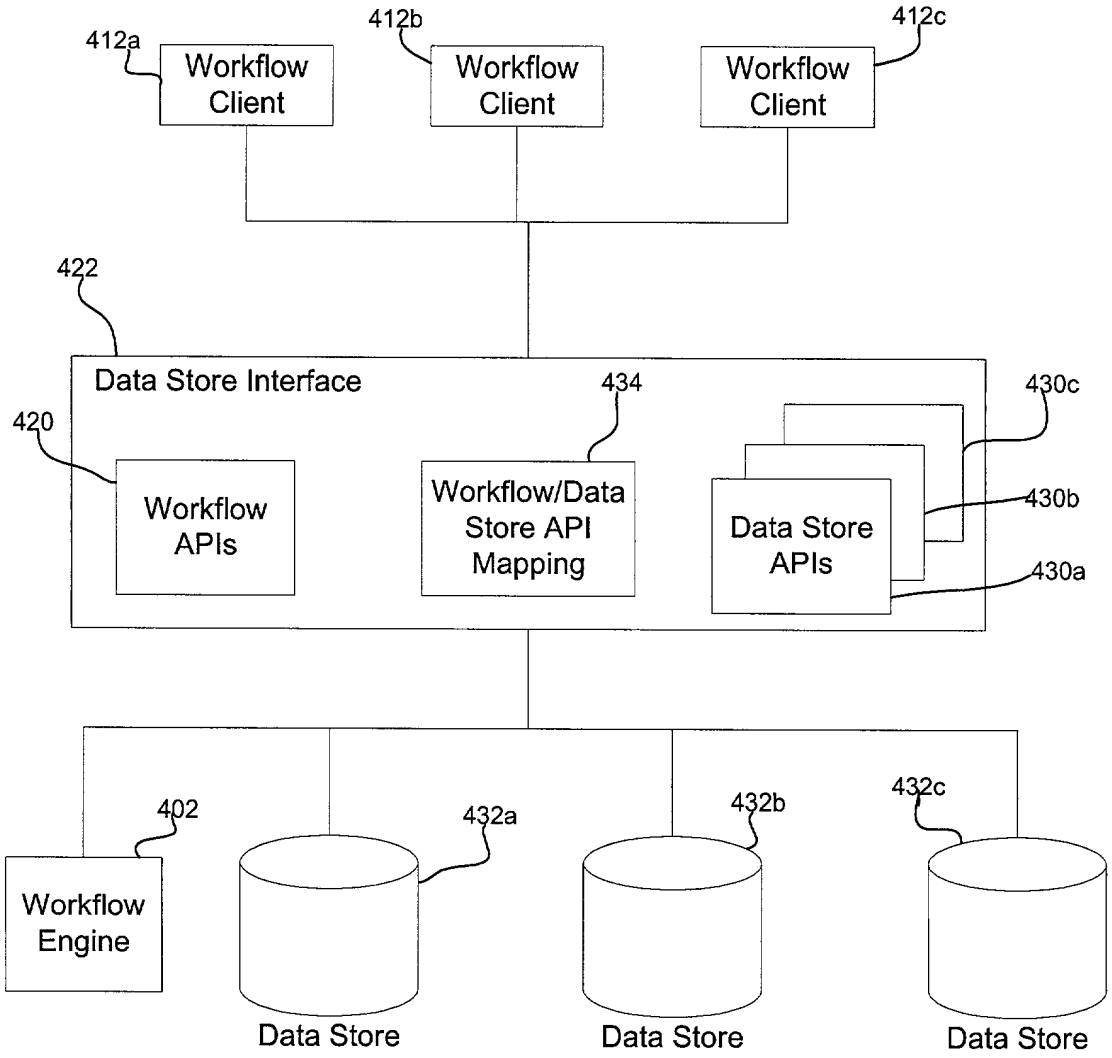

412a
Workflow Client
412b
Workflow Client
412c
Workflow Client
422
Data Store Interface
420
Workflow APIs
434
Workflow/Data Store API Mapping
430c
430b
Data Store APIs
430a
402
Workflow Engine
432a
Data Store
432b
Data Store
432c
Data Store

FIG. 11

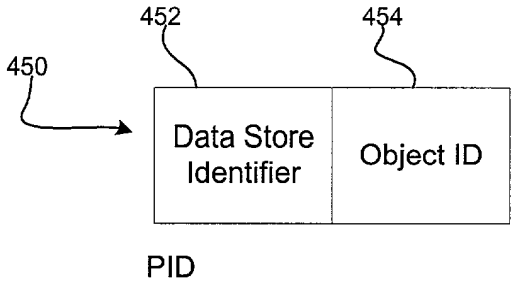


FIG. 12

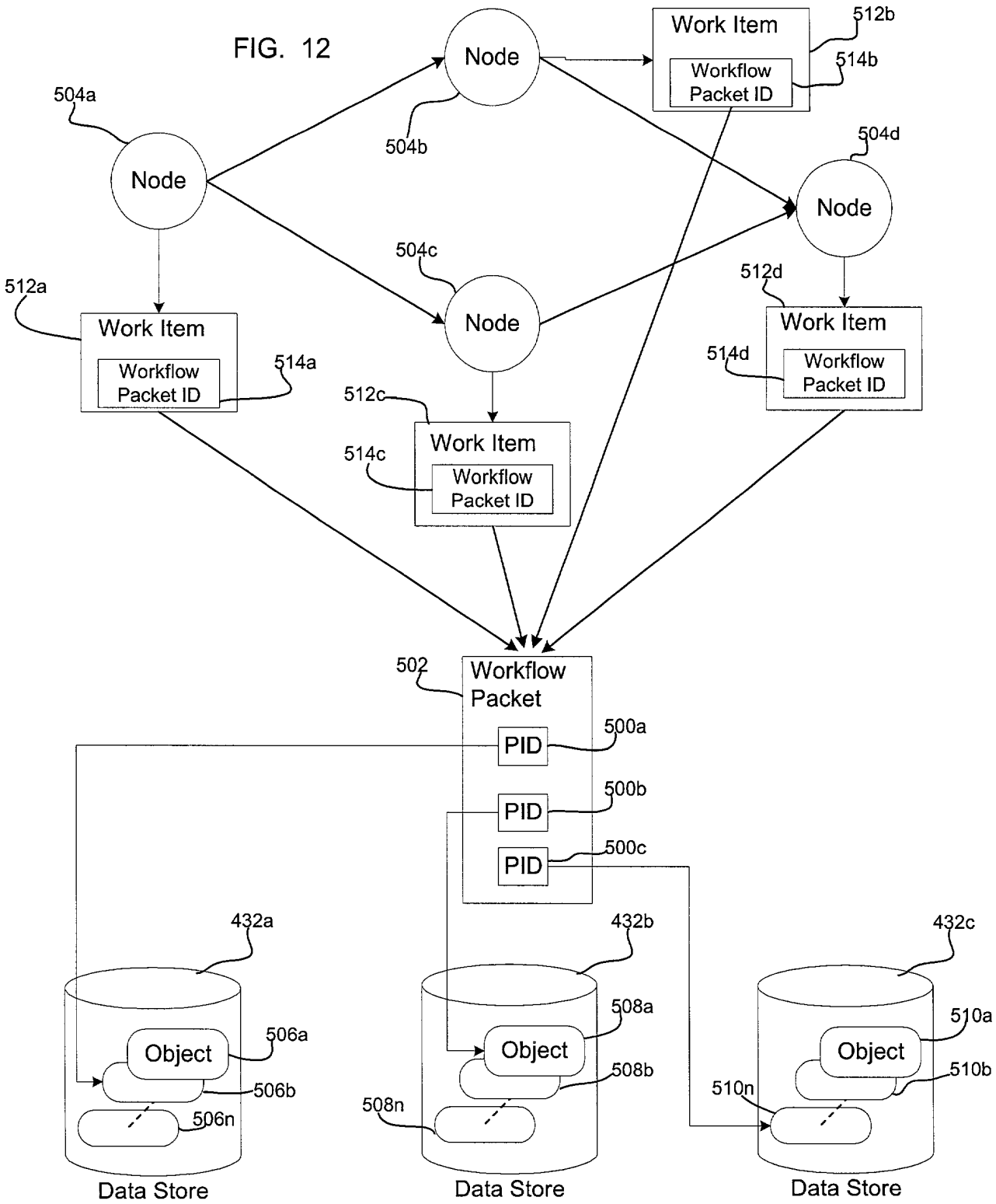

Node
Work Item
512b
Workflow Packet ID
514b
504a
504b
504d
Node
Node
504c
512a
512d
Work Item
Workflow Packet ID
514a
514d
512c
Work Item
514c
Workflow Packet ID
502
Workflow Packet
PID
500a
PID
500b
PID
500c
432a
432b
432c
Object
506a
506b
506n
508a
508b
508n
510a
510b
510n
Data Store
Data Store
Data Store

FIG. 13

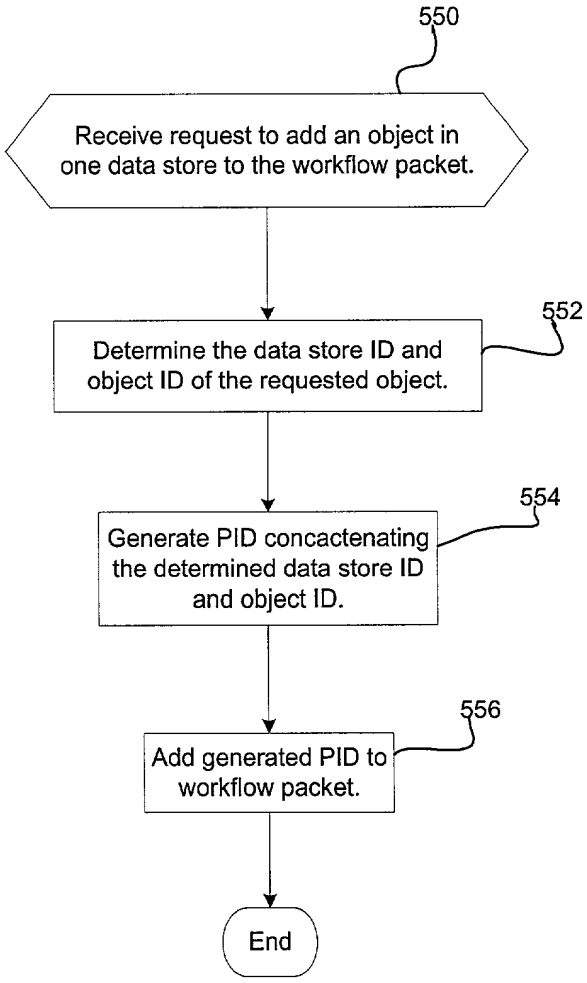
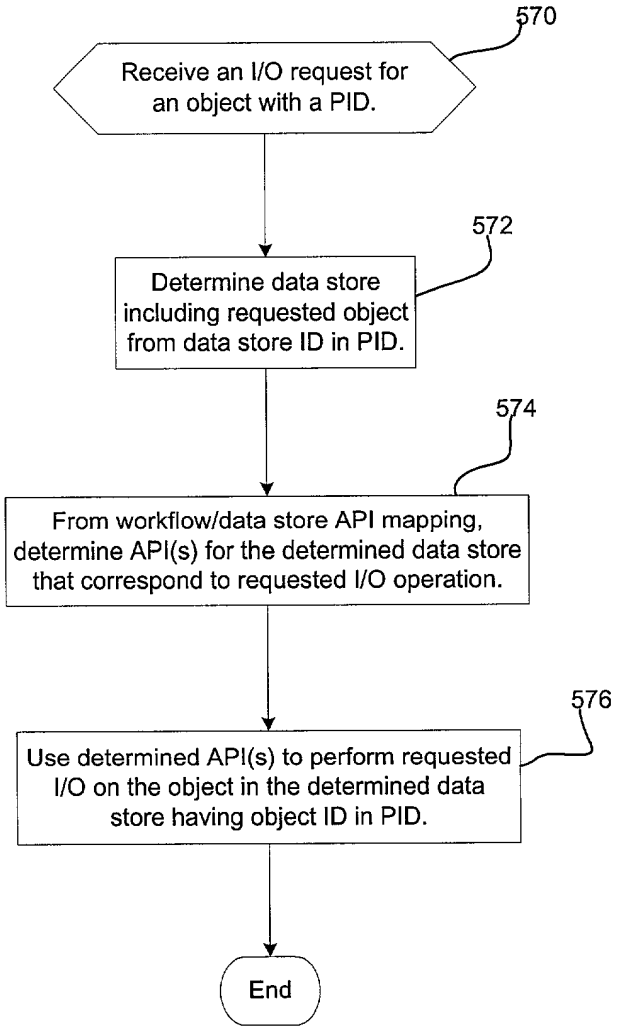

550
Receive request to add an object in one data store to the workflow packet.
552
Determine the data store ID and object ID of the requested object.
554
Generate PID concactenating the determined data store ID and object ID.
556
Add generated PID to workflow packet.
End

FIG. 14


570
Receive an I/O request for an object with a PID.
572
Determine data store including requested object from data store ID in PID.
574
From workflow/data store API mapping, determine API(s) for the determined data store that correspond to requested I/O operation.
576
Use determined API(s) to perform requested I/O on the object in the determined data store having object ID in PID.
End